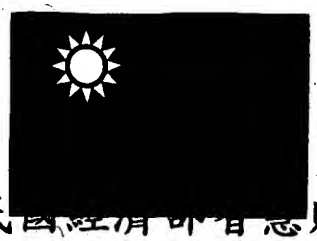
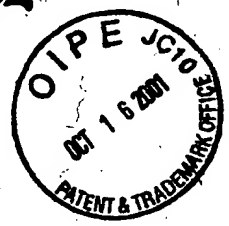
USSN 09/910,829





中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this office of the application as originally filed which is identified hereunder:

申　請　日：西元 2000 年 12 月 21 日
Application Date

申 請 案 號：089127525
Application No.

申　請　人：凌航科技股份有限公司
Applicant(s)

CERTIFIED COPY OF PRIORITY DOCUMENT

局　長
Director General

陳明邦

發文日期：西元 2001 年 8
Issue Date

發文字號：09011011943
Serial No.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Chun-Yang CHENG, et al.)
)
Serial No.: 09/916,829)
)
Filed: July 26, 2001) Our Ref: B-4251 618951-9
)
For: "MODULAR MULTIPLIER AND AN)
ENCRYPTION/DECRYPTION PROCESSOR)
USING THE MODULAR MULTIPLIER") Date: October 10, 2001

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

Commissioner of Patents and Trademarks
Box New Patent Application
Washington, D.C. 20231

Sir:

[X] Applicants hereby make a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
Taiwan, R.O.C.	21 December 2000	89127525

[ ] A certified copy of each of the above-noted patent application was filed with the Parent Application No._____.

[X] To support applicants' claim, a certified copy of the above-identified foreign patent application is enclosed herewith.

[ ] The priority document will be forwarded to the Patent Office when required or prior to issuance.

Respectfully submitted,

Richard P. Berg
Attorney for Applicant
Reg. No. 28,145

LADAS & PARRY
5670 Wilshire Boulevard
Suite 2100
Los Angeles, CA 90036
Telephone: (323) 934-2300
Telefax: (323) 934-0202

申請日期	
案　　號	
類　　別	

A4
C4

（以上各欄由本局填註）

發明 新型 專利說明書

一、發明新型名稱	中　文	模乘法器和使用此模乘法器之加/解密器
	英　文	
二、發明創作人	姓　名	1.鄭俊揚　　2.蔡維昌
	國　籍	中華民國
	住、居所	1.新竹市新莊里 11 鄰新莊街 142 號 7 樓 2.台北縣板橋市吉翠里 15 鄰裕民路 149 號 5 樓
三、申請人	姓　名（名稱）	凌航科技股份有限公司
	國　籍	中華民國
	住、居所（事務所）	新竹科學工業園區展業二路十八路六樓
	代表人姓　名	許仁旭

C5
D5

四、中文發明摘要（發明之名稱：模乘法器和使用此模乘法器之加/解密器）

一種模乘法器以及使用此模乘法器之加/解密器，主要係應用於晶片面積較小並且快速運算的應用上。在此模乘法器中主要係用來實現 Montgomery 演算法，並且將運算元拆解成固定長度的位元資料，再反覆計算而產生所需要的結果。在演算法中包含遞迴架構的兩層先乘法後加法的運算。透過多工器的選擇，可以利用單一資料路徑在不同時間上計算出所需的模乘法結果。

英文發明摘要（發明之名稱：　　）

（請先閱讀背面之注意事項再填寫本頁各欄）

裝

訂

線

五、發明說明（ 1 ）

本發明係有關於一種模乘法(modular multiplication)運算器架構，特別是根據高基數(high-radix)Montgomery 運算法所實現的模乘法器，能夠節省所需要的晶片實作面積同時達到高速運算的目的。

由於資料傳輸網路化和資料數位化的需要，目前針對資料提供安全機制的密碼技術已成為最重要的課題之一。其基本原理是，明文(plaintext)資料透過一加密機制(encryption)以及使用者所選擇的加密金鑰(encryption key)，可以轉換為一密文(ciphertext)資料。在資料傳輸或保存期間則是透過密文形式達成，當傳輸接收端接收到此密文或是需要解開此密文時，則是透過一對應的解密機制(decryption)和對應的解密金鑰(decryption key)還原出明文資料。由於資料傳輸過程中或是資料保存時係採用密文形式保存，因此在不知道對應解密金鑰的情況下，便可以達到資料的安全性。

在一般密碼系統的安全機制下，其安全性基本上是建立在金鑰是否易於解出或取得的基礎上。換言之，根據現有資料推論出金鑰的難易程度便成為密碼系統的安全性指標。目前所採用的密碼系統，主要可以區分為私密金鑰密碼系統(private key cryptosystem)以及公開金鑰密碼系統(public key cryptosystem)兩大類型。在私密金鑰系統中，加密金鑰和解密金鑰是相同的，例如目前使用最多的 DES 系統。在公開金鑰密碼系統中，由於加密金鑰和解密金鑰相同意味著必須存在一絕對安全的密碼傳輸路徑，才能夠保

五、發明說明（2）

證密碼系統的安全性，這也是私密金鑰密碼系統最主要的缺點之一。另一方面，公開金鑰密碼系統則沒有這方面的問題。在公開金鑰密碼系統中，加密金鑰和解密金鑰是不同的。在一對加密金鑰和解密金鑰裡，加密金鑰是可以公開的金鑰，利用此加密金鑰對明文加密成密文後，則只有對應的解密金鑰才可以還原成明文。另外，此系統同樣必須保證在理論上無法或很難由公開金鑰推論出解密金鑰，藉以確保系統的安全性。此類加密系統的代表者即為RSA(Rivest，Shamir，Adleman)。由於公開金鑰密碼系統沒有金鑰傳遞時所造成的問題，也不會有金鑰管理上的問題，因此目前已逐漸成為密碼系統的主流。另外，其中的解密金鑰還可以做到數位簽章(digital signature)的驗證功能。

RSA 密碼系統主要是利用模指數(modular exponential)運算來進行加/解密的動作，其加/解密動作可以表示如下：

$$C=M^{E}(\text{mod } N) \qquad (1)$$

$$M=C^{D}(\text{mod } N) \qquad (2)$$

其中 N=PQ 並且 $ED\equiv 1 \text{ mod}(P-1)(Q-1)$。其中 M 表示明文，C 表示密文，而 E、D 則分別表示加密金鑰和解密金鑰。上述參數 N 係為兩個質數 P、Q 的乘積。第(1)式表示加密的動作，亦即明文 M 利用模乘法運算(E、N)得到密文 C。第(2)式則表示解密的動作，亦即密文 C 利用模乘法運算(D、N)還原得到明文 M。在 RSA 密碼系統中，模指數運算是非常複雜且計算時間非常耗時的步驟，因此目前大多數是利用模乘法運算來實現模指數運算，特別是利用 Montgomery

五、發明說明（ 3 ）

演算法來計算兩數值模乘法運算的結果。以下首先說明 Montgomery 演算法針對 AB(mod N)的基本運作方式：

<演算法 1>

首先設 $R_0=0$；

迴圈 i=0~n-1，執行下列動作：

$$q_i=R_i+a_iB \pmod 2 \qquad (3)$$

$$R_{i+1}=(R_i+a_iB+q_iN)/2 \qquad (4)$$

其中 $A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots\cdots + a_0$；$B = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots\cdots + b_0$，且 a_i、b_i、$q_i \in \{0、1\}$。

上述的演算法需要執行 n 個迴圈(loop)，使用到 n 位元的加法器以及 1 乘 n 的乘法器。將每個迴圈所得到的算式分別乘上 2^0、2^1、2^2、…、2^{n-1} 後相加，可以得到以下結果：

$$2^nR_n \equiv AB \pmod N \qquad (5)$$

實際的運算結果則是 R_n，根據第(5)式可以表示為：

$$R_n \equiv 2^{-n}AB \pmod N \qquad (6)$$

要利用上述 Montgomery 演算法來計算在第(1)式或是第(2)式的模指數運算時，則可以依據下列預運算、指數運算以及後段運算三個步驟來完成：

$$MGM(M, 2^{2n}) \equiv 2^nM \pmod N \qquad (7)$$

$$MGM(2^nM^a, 2^nM^b) \equiv 2^nM^{a+b} \pmod N \qquad (8)$$

$$MGM(2^nM^E, 1) \equiv M^E \pmod N \qquad (9)$$

其中 MGM(・,・)表示上述 Montgomery 演算法計算結

五、發明說明（ 4 ）

果的 R_n，亦即第(6)式所表示的 $2^{-n}AB \pmod N$。

在演算法 1 由於需要執行 n 次的迴圈，因此在運算上比較耗時。因此，利用高基數(high radix，基數 2^k)Montgomery 演算法可以有效地利用晶片面積來增加其運算速度。在此演算法中，主要是將運算元 A 拆成$\lceil n/k \rceil$組，每組 k 位元，讓編碼或解碼時計算一次模乘法運算的迴圈數降為 n/k 次，達到增加速度的目的，其演算法可以表示如下：

<演算法 2>

首先設 $R_0=0$；

迴圈 $i=0 \sim \lceil n/k \rceil -1$，執行下列動作：

$$q_i = (R_i+a_iB)*N_1 \pmod{2^k} \qquad (10)$$

$$R_{i+1} = (R_i+a_iB+q_iN)/2^k \qquad (11)$$

其中 N_1 滿足 $N*N_1 \equiv -1 \pmod{2^k}$，$A = a_{\lceil n/k \rceil -1}(2^k)^{\lceil n/k \rceil -1} + a_{\lceil n/k \rceil -2}(2^k)^{\lceil n/k \rceil -2} + \cdots\cdots + a_0*(2^k)^0$，且 a_i、$q_i \in \{0、1、2、\cdots、2^k-1\}$，k 為大於 0 的整數。

在演算法 2 中，雖然已然達到減少迴圈數量的目的，不過其仍然可以簡化，亦即將運算元 B 位移 k 位元以及改變為 N_2 的參數值，以便將第(10)式中的乘法和加法運算去除掉，因此整個演算法即變為：

<演算法 3>

首先設 $R_0=0$；

五、發明說明（ 5 ）

迴圈 i=0~⌈n/k⌉，執行下列動作：

$$q_i = R_i(\text{mod } 2^k) \quad (12)$$

$$R_{i+1} = (R_i+q_i*N_2)/2^k+a_iB \quad (13)$$

其中 $N_2=mN\equiv-1(\text{mod } 2^k)$.

同樣地，將每個迴圈所得到的算式分別乘上 2^0、2^1、2^2、…、2^{n-1} 後相加，可以得到以下結果：

$$2^{n+k}R_{(n/k)+1}\equiv A*2^k*B + Q*N_2 \quad (14)$$

換言之，演算法 3 所得到的最後結果 $R_{(n/k)+1}$ 確實是滿足如第(5)所示之關係，亦即：

$$2^nR_{(n/k)+1}\equiv AB \text{ (mod } N) \quad (15)$$

演算法 3 最大的優點在於其可以利用相同的運算架構，亦即一次乘法和一次加法，來處理演算法中第(13)式計算 R_{i+1} 的動作。首先，假設在第(13)式中部分算式為：

$$X= R_i+q_i*N_2 \quad (16)$$

因此，第(13)式可以改寫為：

$$R_{i+1}=X/2^k + a_i*B \quad (17)$$

再令 $Y=X/2^k$，第(17)又可以改寫為：

$$R_{i+1}=Y+ a_i*B \quad (18)$$

在第(16)式和第(18)式中，同樣都是先做一次乘法運算再做一次加法運算，而且相對應的運算元具有相同的位元數。換言之，可以利用同一套資料路徑(data path)在不同時間做運算的處理，如此便可以節省晶片所需要的面積。

在習知 Montgomery 演算法 3 中，同樣有其實作上的問題。回到第(16)式和第(18)式所揭露的運算中，其中的乘法

五、發明說明（ 6 ）

運算是需要一個 k 乘 n 的乘法器。當 n、k 較大時，乘法器面積就相當可觀，例如當 k=32 而 n=1024 時。當所需要的乘法器過大時，自然會產生晶片面積增加的現象。此一現象對於一般尺寸的晶片基本上影響不致於太大，但是對於晶片面積要求較嚴格的應用，例如智慧卡等等，則會造成嚴重的影響。因此，本發明即是根據此問題，對於高基數 Montgomery 演算法提出改進之方案，以達到降低晶片面積並且達到高速運算的目的。

有鑑於此，本發明提出一種模乘法器以及使用此模乘法器之加/解密器，能夠降低晶片所需要之面積，並且達到高速運算的目的。

根據上述之目的，本發明提供一種模乘法器，用以處理一第一運算元和一第二運算元相對於一模數進行模乘法運算，其包含一指令，上述指令具有內部遞迴之內部乘加法運算和外部乘加法運算。其中包括一第一暫存裝置，用以儲存上述第一運算元，上述第一運算元係拆成具有固定長度之複數第一子段落；一第二暫存裝置，用以儲存上述第二運算元，上述第二運算元係拆成具有固定長度之複數第二子段落；一第三暫存裝置，用以儲存模乘法運算之參數；一多工裝置，耦接於上述第一暫存裝置、上述第二暫存裝置和上述第三暫存裝置，用以依據內部乘加法運算和外部乘加法運算之需要，依序由上述第一子段落、上述第二子段落和上述參數中選擇第一乘法運算元和第二乘法運算元；一乘法裝置，耦接於上述多工裝置，用以計算上述

五、發明說明（ 7 ）

第一乘法運算元和上述第二乘法運算元之乘積；以及一加法裝置，耦接於上述乘法裝置，在上述內部乘加法運算期間，根據上述乘積產生一中間值，在上述外部乘加法運算期間，根據上述乘積值和上述中間值，產生上述模乘法運算結果。

其中，上述模乘法器可以一種加密器或解密器，例如RSA 密碼器，其中利用加密/解密金鑰進行加密/解密動作的模指數運算，可以透過上述之模乘法器來實現。而這樣的加密器和解密器，還可以應於例如智慧卡等等的應用上，尤其是其需要晶片面積較小同時能夠快速運算，上述模乘法器則可以解決上述的問題。

圖式簡單說明

第 1 圖表示本發明實施例之模乘法運算器的方塊圖。

第 2 圖表示在本發明實施中第一個次迴圈內加法器運算時動作的示意圖。

第 3 圖表示在本發明實施中第二個次迴圈內加法器運算時動作的示意圖。

第 4 圖表示利用本發明實施例中模乘法器實現 RSA 加/解密器的方塊示意圖。

第 5 圖表示利用本發明實施例所示之加/解密架構使用於智慧卡時之示意圖。

[符號說明]

五、發明說明（ 8 ）

101、102、103、104、105~暫存器；201、202~多工器；203~乘法器；204~控制單元；301、302、303、305、306~正反器；304~加法器；10~RSA 加/解密器；12~加/解密核心；14~模乘法器核心；20~智慧卡；22~通訊界面；24~加/解密器；26~內部記憶體。

實施例

本實施例中的模乘法運算器主要是為了改善習知技術中晶片面積過大的問題所提出的一種方案。在習知技術的演算法 3 中需要一個 k 乘 n 的乘法器，因此需要大量的晶片面積來實現。以下，則先說明本實施例之演算法內容，再說明其對應之模乘法運算器架構。

首先處理習知技術的演算法 3 中的乘法運算部分。為了降低乘法器的晶片面積，可以將原來乘法運算元中具有 n 位元的部分(亦即第(16)式的 N_2 和第(18)式的 B)拆成 n/k 組，每組 k 位元再計算，亦即：

<演算法 4>

首先設 $R_0=0$；

迴圈 $i=0\sim\lceil n/k\rceil$，執行下列動作：

$$q_i = R_i \pmod{2^k} \qquad (19)$$

迴圈 $j=0\sim\lceil n/k\rceil-1$，執行下列動作：

$$(R_{i+1})_j = ((R_i)_j+q_i*(N_2)_j)/2^k+a_iB_j \qquad (20)$$

五、發明說明（ 9 ）

因此，其中的乘法運算 $q_i*(N_2)_j$ 和 a_i*B_j 均是 k 乘 k 的乘法運算，所以乘法器晶片面積可以大幅縮小，但是其中 j 迴圈需要處理進位以及累加的運算。以上演算法可以利用更具體的方式呈現如下：

<演算法 5>

首先設 $R_0=0$；

迴圈 $i=0\sim\lceil n/k\rceil$，執行以下動作：

$$q_i=R_i \bmod 2^k \quad (21)$$

$$W=q_i*(N_2)_0 \quad (22)$$

$$C_{-1}=(R_i)_0+W[(k-1):0]\text{的溢位輸出} \quad (23)$$

$$V=0 \quad (24)$$

迴圈 $j=0\sim\lceil n/k\rceil-1$，執行以下動作：

$$Z=W \quad (25)$$

$$W=q_i*(N_2)_{j+1} \quad (26)$$

$$U=a_i*B_j \quad (27)$$

$$\{C_j, (R_{i+1})_j\}=(R_i)_{j+1} + W[(k-1):0] + Z[(2k-1):k] + U[(k-1):0]+V[(2k-1):k]+C_{j-1} \quad (28)$$

$$V=U \quad (29)$$

其中符號 W、Z、U、V 係表示臨時用暫存器(temporary register)，另外 C_j 表示進位輸出位元，而 $\{ C_j, (R_{i+1})_j \}$ 則是表示 k 位元加法運算的總和。再者，藉由選擇適當的 q_i、N_2，可以使得 $(R_i)_0+W[(k-1):0]$ 為 0。

演算法 5 中需要兩個 k 乘 k 的乘法器，來計算第(26)式

五、發明說明（10 ）

和第(27)式中的 $q_i*(N_2)_{j+1}$ 以及 a_i*B_j。事實上，演算法 5 在 j 迴圈中的運算是可以再進一步拆成兩個次迴圈的運算，在以下的演算法 6 中，即拆成第一個次迴圈處理原 j 迴圈的 $((R_i)_j+q_i*(N_2)_j)/2^k$，而第二個次迴圈則處理其餘的運算部分。演算法來可以表示為：

<演算法 6>

首先設 $R_0=0$；

迴圈 $i=0\sim\lceil n/k\rceil$，執行下列動作：

$$q_i = R_i(\text{mod } 2^k) \qquad (30)$$

第一個次迴圈 $j=0\sim\lceil n/k\rceil-1$，執行下列動作：

$$Y_j = ((R_i)_j+q_i*(N_2)_j)/2^k \qquad (31)$$

第二個次迴圈 $j=0\sim\lceil n/k\rceil-1$，執行下列動作：

$$(R_{i+1})_j=Y_j+a_i*B_j \qquad (32)$$

同樣地，可以將演算法 6 以更具體的方式呈現：

<演算法 7>

首先設 $R_0=0$；

迴圈 $i=0\sim\lceil n/k\rceil$，執行以下動作：

$$q_i=R_i \bmod 2^k \qquad (33)$$

$$W=q_i*(N_2)_0 \qquad (34)$$

$C_{-1}=(R_i)_0+W[(k-1):0]$的溢位輸出 (35)

第 1 個次迴圈 $j=0\sim\lceil n/k\rceil-1$，執行以下動作：

$$Z=W \qquad (36)$$

五、發明說明（11）

$$W=q_i*(N_2)_{j+1} \quad (37)$$

$$\{C_j, Y_j\}=(R_i)_{j+1} + W[(k-1):0] + Z[(2k-1):k] + C_{j-1} \quad (38)$$

$$C_{-1}=0 \quad (39)$$

$$Z=0 \quad (40)$$

第 2 個次迴圈 $j=0\sim\lceil n/k\rceil-1$，執行以下動作：

$$W=a_i*B_j \quad (41)$$

$$\{C_j, (R_{i+1})_j\}=Y_j + W[(k-1):0] + Z[(2k-1):k] + C_{j-1} \quad (42)$$

$$Z=W \quad (43)$$

在演算法 6 和演算法 7 中，係將原來演算法 5 的 j 迴圈拆成兩個次迴圈，此架構可以讓原本需要 2 個 k 乘 k 的乘法器縮減成一個，減少所需要的晶片面積。另一方面，利用上述演算法執行的模乘法運算實際上的速度也非常快。以 n=1024，k=32 為例，上述演算法需要一個 32× 32 的乘法器以及數個加法器。若僅以要耗時的乘法運算來計算時序，假設執行一個 32× 32 的乘法需要一個時序，那麼執行上述演算法中第一個次迴圈的第(31)式需要用掉(1024/32)=32 個次序，同樣的，執行第二個次迴圈的第(32)式也需要用掉 32 個時序。因此，整個模乘法運算(亦即 i 迴圈)則需要用掉(1024/32+1) × (32+32)=2112 個次序。另一方面，如果整個 1024 位元的 RSA 編碼或解碼模指數運算是採用 H 演算法(H-Algorithm)，則整個電路最多需要使用到 2

五、發明說明（12 ）

× 2112× 1024 個時序，相當於大約 4M 個時序，如果以 n、k 為參數表示時，則所需的時序大約是 $4n^2(n+1)/k^2$。由此可知，上述演算法中不僅可以將晶片面積大幅縮小，同時也可以達到高速運算的目的。

第 1 圖表示本發明實施例中運算演算法 6 或演算法 7 所設計之模乘法運算器之方塊圖。第 1 圖所示之模乘法運算器架構中，主要係根據上述演算法 7 所示之各種運算格式加以設計，其主要包括下列包括暫存器 101、暫存器 102、暫存器 103、暫存器 104、暫存器 105；多工器 201、多工器 202、乘法器 203、控制單元 204；以及正反器 301、正反器 302、正反器 303、加法器 304、正反器 305 和正反器 306。以下分別描述各元件之基本功能。

暫存器 101 是用來儲存 Montgomery 演算法之結果$(R_{i+1})_j$或者是第一個次迴圈中的中間運算元 Y_j。暫存器 102~105 主要是用來分別儲存在演算法 7 中兩個乘法運算式(即第(37)式和第(41)式)中的運算元，也就是 A、N_2、B、q_i。其中，N_2、A、B 係為固定常數，而 a_i 是運算元 A 在第 i 迴圈的部分位元，$(N_2)_j$ 以及 B_j 則是運算元 N_2 和 B 在第 j 迴圈的部分位元。另外，根據第(33)式，暫存器 105 所儲存的 q_i 係為 R_i 除以 2^k 所得到的餘數，也就是 R_i 中位元(k-1)~0 位元的部分。因此，只要將暫存器 101 中所儲存的 R_i 中取出較低的 k 位元，便可以產生暫存器 105 中的 q_i。

多工器 201 和多工器 202 則是用來切換不同次迴圈中乘法運算所需要的運算元，例如在第一個次迴圈中，第(37)

五、發明說明（13 ）

式的乘法運算需要 q_i 和 $(N_2)_j$，而在第二個次迴圈中，第(41)式的乘法運算需要 a_i 和 B_j。多工器 201 和多工器 202 則是透過控制單元 204 的控制信號 CTRL 加以切換。兩者之輸出則是由 k 乘 k 的乘法器 203 執行乘法運算，產生長度 2k 的乘積(亦即 W)。

正反器 301~303 則是利用儲存乘法器的結果，並且輸入到加法器 304 中進行第(38)式和第(42)式的加法運算。長度為 2k 的 W 暫存值則拆成兩個長度 k 的資料，其中低位元 W[(k-1):0]係送到正反器 302，高位元 W[(2k-1):k]則送到正反器 301。而正反器 303 則是儲存前一乘法結果中的高位元 Z[(2k-1):k]。正反器 305 則是儲存前一加法運算結果中的溢位位元 C_{j-1}。加法器 304 係執行第一個次迴圈中第(38)式或是第二個次迴圈中第(42)式的加法運算，兩者的差別在於使用 $(R_i)_{j+1}$ 或是 Y_j。在以下的運算中，當執行第一個次迴圈時正反器 306 所儲存的是 Y_j，則會暫時回存到暫存器 101 中，而在第二個次迴圈時正反器 306 所儲存的是 $(R_i)_{j+1}$，亦儲存到暫存器 101 中。

上述簡單說明了各元件的作用。第 1 圖所示之模乘法器的操作則詳細說明如下：

根據演算法 7 所示，每次 i 迴圈開始的第一道指令即是取 R_i 除以 2^k 的餘數，也就是取暫存器 101 中的 R_i 較低 k 位元，儲存到暫存器 105 中。

接著進入第一個次迴圈，其係利用 q_i、$(N_2)_j$ 和 $(R_i)_j$ 等等參數來計算出 Y_j。首先，在第一個次迴圈中，q_i 值是不

五、發明說明（14 ）

會改變的(因為其下標為 i)，因此當 q_i 值被計算出來後，便可以儲存在暫存器 105 中。而隨著 j 值的變化，暫存器 103 中則送出對應的$(N_2)_j$。每次利用乘法器 203 完成乘法動作後，其乘積的較高 k 位元 W[(2k-1):k]被送到正反器 301，而較低 k 位元 W[(k-1):0]則被送到正反器 302。被送到正反器 301 的較高 k 位元 W[(2k-1):k]則會延遲一個時序，在計算下一組 Y_{j+1} 時才會計入。另一方面，被送到正反器 302 的較低 k 位元 W[(k-1):0]則是與前一次乘積的較高 k 位元 Z[(2k-1):k](儲存於正反器 303)、$(R_i)_{j+1}$(儲存於暫存器 101)，以及前一次加法運算中的溢位位元 C_{j-1}(儲存於正反器 305)，透過加法器 304 產生 Y_j。並且在下一個時序中，儲存到暫存器 101 中。

第 2 圖表示第一個次迴圈中加法器運算時動作的示意圖，其中假設 k=32，而 n=1024。其中第 1 行表示第(35)式的計算動作。當 j=0 時，加法器 304 則將 R_i[63:32]、$(q_i(N_2)_1)$[31:0]、$(q_i(N_2)_0)$[63:32]以及溢位位元 C_{-1} 相加，產生 Y[31:0]。當 j=1 時，加法器 304 則將 R_i[95:64]、$(q_i(N_2)_2)$[31:0]、$(q_i(N_2)_1)$[63:32]以及溢位位元 C_0 相加，產生 Y[63:32]。其餘依此類推。直到 j=31 時，則可以計算出 Y[1023:992]。此時，Y[1023:0]便計算完成。

接著進入第二個次迴圈，其係利用 a_i、B_j 和 Y_j 等等參數來計算出$(R_{i+1})_j$。同樣地，a_i 值是不會改變的(因為其下標為 i)，並且由暫存器 102 送出。而隨著 j 值的變化，暫存器 104 中則送出對應的 B_j。每次利用乘法器 203 完成乘法動作

五、發明說明（15）

後，其乘積的較高 k 位元 W[(2k-1):k]被送到正反器 301，而較低 k 位元 W[(k-1):0]則被送到正反器 302。被送到正反器 301 的較高 k 位元 W[(2k-1):k]則會延遲一個時序，在計算下一組$(R_{i+1})_j$時才會計入。另一方面，被送到正反器 302 的較低 k 位元 W[(k-1):0]則是與前一次乘積的較高 k 位元 Z[(2k-1):k](儲存於正反器 303)、Y_j(儲存於暫存器 101)，以及前一次加法運算中的溢位位元 C_{j-1}(儲存於正反器 305)，因此其動作與上個次迴圈是相同的。所產生的$(R_{i+1})_j$則在下一個時序中，儲存到暫存器 101 中。

第 3 圖表示第二個次迴圈中加法器運算時動作的示意圖，配合第 2 圖所示的情況。當 j=0 時，加法器 304 則將 Y[31:0]和(a_iB_0)[31:0]相加，產生 R_{i+1}[31:0]，此時上次乘積的較高 k 位元尚不存在。當 j=1 時，加法器 304 則將 Y[63:32]、(a_iB_1)[31:0]以及(a_iB_0)[63:32]相加，產生 R_{i+1}[63:32]。其餘依此類推。直到 j=31 時，則可以計算出 R_{i+1}[1023:992]。此時，R_{i+1}[1023:0]便計算完成。

重複針對不同的 i 值計算其對應的 R_i 值。最後，便可以取得 Montgomery 演算法的最後結果，其代表 $2^{-n}AB(\text{mod } N)$ 的模乘法。必須注意的是，在上述第一個次迴圈和第二個次迴圈之間，相關正反器內容資料必須加以清除，以便利用相同資料路徑計算不同的算式。控制單元 204 的控制信號 CTRL，則可以控制整個運算的進行方式，亦即，利用切換不同的乘法運算元，來依序計算出演算法 6 或演算法 7 所需要的各種計算動作。

五、發明說明（16 ）

如上所示，本發明實施例所揭露之模乘法器，其優點即在於可以節省晶片面積同時能夠快速地執行運算。第 4 圖表示利用本發明實施例中模乘法器實現 RSA 加/解密器的方塊示意圖。如圖所示，RSA 加/解密器 10 包括加/解密核心 12 以及模乘法器核心 14。其中，模乘法器核心 14 可以利用如第 1 圖所示之架構實現，亦即對於 A、B 兩個運算元計算其模乘法結果。加/解密核心 12 則是利用第(7)式、第(8)式、第(9)式所述之預運算、指數運算以及後段運算三步驟，協同模乘法器核心 14 完成 RSA 加/解密所需要的模指數運算，藉以將明文 M 加密成密文 C，或者是將密文 C 還原成明文 M。

第 5 圖則表示利用本發明實施例所示之加/解密架構，使用於智慧卡時之示意圖。智慧卡由於受限於規格以及易於攜帶等等的條件，通常對於晶片面積的要求較嚴格。在第 5 圖所示之智慧卡 20 中，則是透過通訊界面 22 與外界交換資料。當資料透過通訊界面智慧卡 20 的內部記憶體 26 時，則必須透過加/解密器 24 進行加密，藉以保障資料的安全性。由於加/解密器 24 需要快速地完成計算並且儘可能佔用較小的面積，因此利用本發明實施例所揭露之模乘法器架構實現時，便可以達到上述之要求。

雖然本發明已以一較佳實施例揭露如上，然其並非用以限定本發明，任何熟習此技藝者，在不脫離本發明之精神和範圍內，當可作些許之更動與潤飾，因此本發明之保護範圍當視後附之申請專利範圍所界定者為準。

六、申請專利範圍

1.一種模乘法器，用以處理一第一運算元和一第二運算元相對於一模數進行模乘法運算，其包含一指令，上述指令具有內部遞迴之內部乘加法運算和外部乘加法運算，其包括：

一第一暫存裝置，用以儲存上述第一運算元，上述第一運算元係拆成具有固定長度之複數第一子段落；

一第二暫存裝置，用以儲存上述第二運算元，上述第二運算元係拆成具有固定長度之複數第二子段落；

一第三暫存裝置，用以儲存模乘法運算之參數；

一多工裝置，耦接於上述第一暫存裝置、上述第二暫存裝置和上述第三暫存裝置，用以依據內部乘加法運算和外部乘加法運算之需要，依序由上述第一子段落、上述第二子段落和上述參數中選擇第一乘法運算元和第二乘法運算元；

一乘法裝置，耦接於上述多工裝置，用以計算上述第一乘法運算元和上述第二乘法運算元之乘積；以及

一加法裝置，耦接於上述乘法裝置，在上述內部乘加法運算期間，根據上述乘積產生一中間值，在上述外部乘加法運算期間，根據上述乘積值和上述中間值，產生上述模乘法運算結果。

2．如申請專利範圍第 1 項所述之模乘法器，其中上述加法運算裝置尚包括：

一第一延遲元件，耦接於上述乘法裝置，用以接收上述乘積之較低半部位元部分；

六、申請專利範圍

一第二延遲元件，耦接於上述乘法裝置，用以接收上述乘積之較高半部位元部分，上述第二延遲元件單元較上述第一延遲元件多一乘法時序；以及

一加法器，耦接於上述第一延遲元件和上述第二延遲元件，並且接收上述中間值，用以執行加法運算。

3．一種加密器，用以根據一模指數運算，利用一加密金鑰對一明文進行加密，上述模指數運算係透過如申請專利範圍第 1 或 2 項所述之模乘法器執行。

4．一種解密器，用以根據一模指數運算，利用一解密金鑰對一密文進行解密，上述模指數運算係透過如申請專利範圍第 1 或 2 項所述之模乘法器執行。

5．一種智慧卡，其包含一加/解密器，用以對於內部資料進行加密和解密之動作，上述加/解密器係根據一模指數運算，利用加密金鑰和解密金鑰進行加密和解密的動作，上述模指數運算係透過如申請專利範圍第 1 或 2 項所述之模乘法器執行。

6.一種模乘法器，用以處理一第一運算元和一第二運算元相對於一模數進行模乘法運算，上述模乘法運算包含一外部迴圈和一內部迴圈，上述內部迴圈包含一指令，上述指令具有內部遞迴之內部乘加法運算和外部乘加法運算，其包括：

一第一暫存裝置，用以儲存上述第一運算元，上述第一運算元係拆成具有固定長度之複數第一子段落，分別對應於上述外部迴圈；

六、申請專利範圍

一第二暫存裝置，用以儲存上述第二運算元，上述第二運算元係拆成具有固定長度之複數第二子段落，分別對應於上述上述內部迴圈；

一第三暫存裝置，用以儲存模乘法運算之第一參數和第二參數；

一多工裝置，耦接於上述第一暫存裝置、上述第二暫存裝置和上述第三暫存裝置，用以依據內部乘加法運算和外部乘加法運算之需要，依序選擇上述第一子段落和上述第一參數、以及上述第二子段落和上述第二參數中之一做為第一乘法運算元和第二乘法運算元；

一乘法裝置，耦接於上述多工裝置，用以計算上述第一乘法運算元和上述第二乘法運算元之乘積；以及

一加法裝置，耦接於上述乘法裝置，在上述內部乘加法運算期間，根據上述乘積產生一中間值，在上述外部乘加法運算期間，根據上述乘積值和上述中間值，產生上述模乘法運算結果；以及

一控制器，用以送出一控制信號，控制上述多工裝置之動作。

7．如申請專利範圍第 6 項所述之模乘法器，其中上述加法運算裝置尚包括：

一第一延遲元件，耦接於上述乘法裝置，用以接收上述乘積之較低半部位元部分；

一第二延遲元件，耦接於上述乘法裝置，用以接收上述乘積之較高半部位元部分，上述第二延遲元件單元較上

六、申請專利範圍

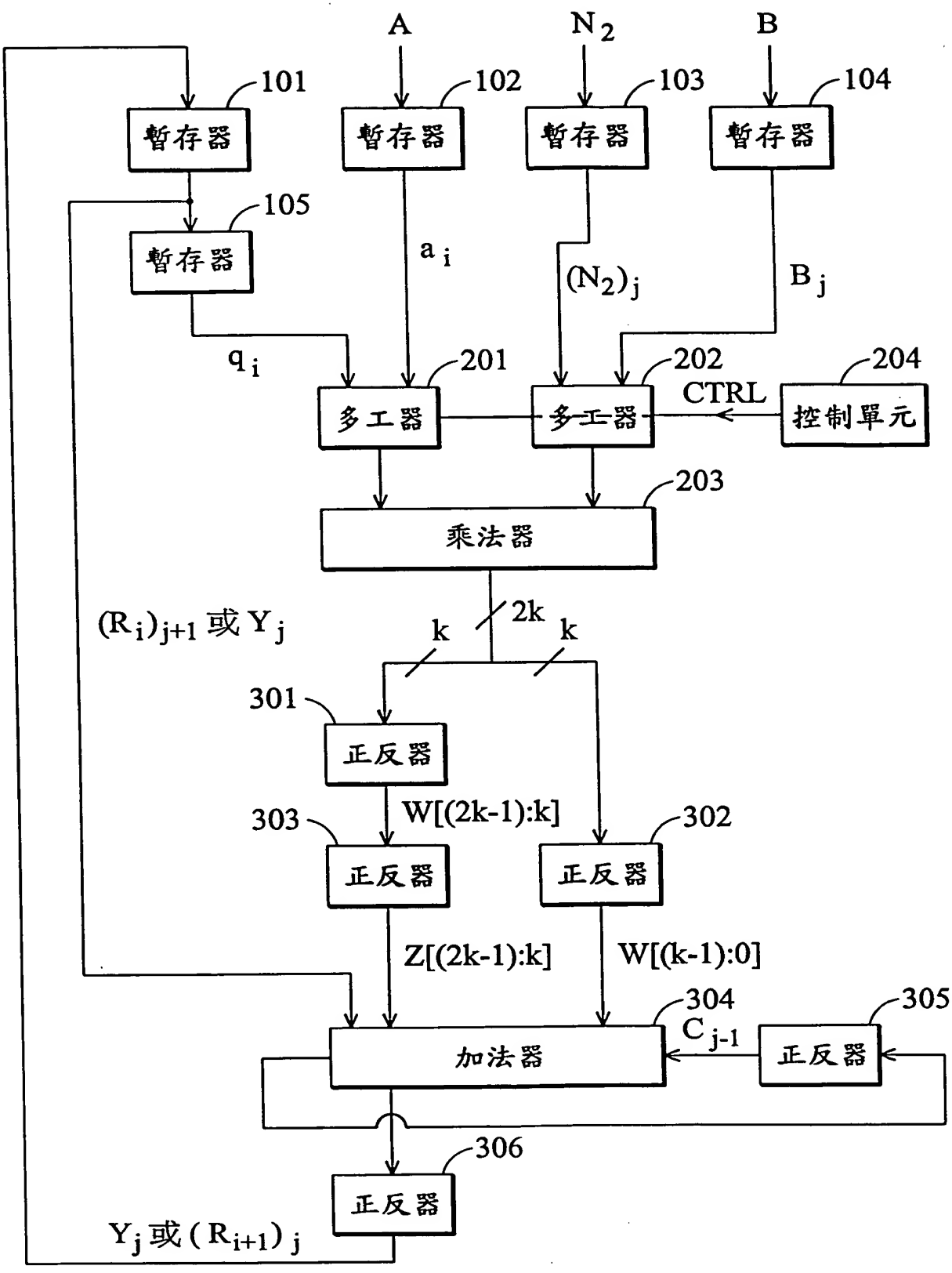
述第一延遲元件多一乘法時序；以及

一加法器，耦接於上述第一延遲元件和上述第二延遲元件，並且接收上述中間值，用以執行加法運算。

8．一種加密器，用以根據一模指數運算，利用一加密金鑰對一明文進行加密，上述模指數運算係透過如申請專利範圍第 6 或 7 項所述之模乘法器執行。

9．一種解密器，用以根據一模指數運算，利用一解密金鑰對一密文進行解密，上述模指數運算係透過如申請專利範圍第 6 或 7 項所述之模乘法器執行。

10．一種智慧卡，其包含一加/解密器，用以對於內部資料進行加密和解密之動作，上述加/解密器係根據一模指數運算，利用加密金鑰和解密金鑰進行加密和解密的動作，上述模指數運算係透過如申請專利範圍第 6 或 7 項所述之模乘法器執行。


A
N2
B
101
102
103
104
暫存器
暫存器
暫存器
暫存器
105
暫存器
ai
(N2)j
Bj
qi
201
202
204
CTRL
多工器
多工器
控制單元
203
乘法器
(Ri)j+1 或 Yj
k
2k
k
301
正反器
W[(2k-1):k]
303
302
正反器
正反器
Z[(2k-1):k]
W[(k-1):0]
304
305
Cj-1
加法器
正反器
306
正反器
Yj 或 (Ri+1)j

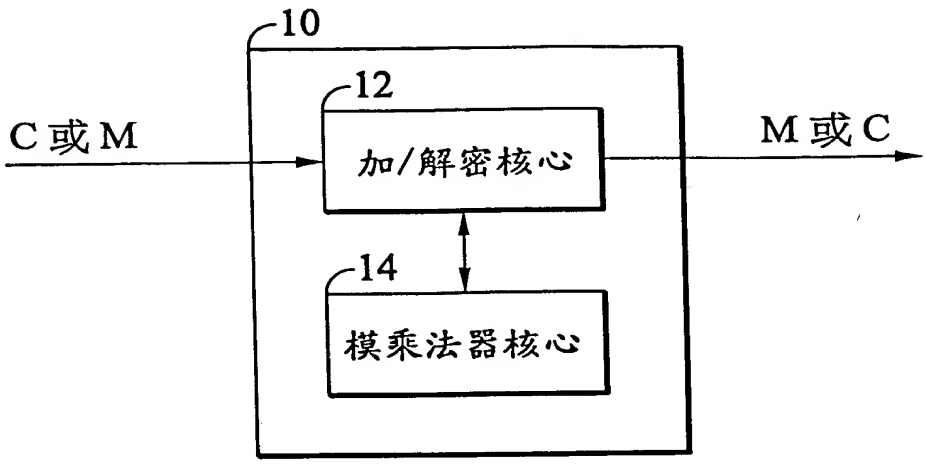
第 1 圖

	j = 0	j = 1	j = 2	• • •
$R_i[31:0]$	$R_i[63:32]$	$R_i[95:64]$	$R_i[127:96]$	• • •
$(q_i(N_2)_0)[31:0]$	$(q_i(N_2)_1)[31:0]$	$(q_i(N_2)_2)[31:0]$	$(q_i(N_2)_3)[31:0]$	• • •
+	$(q_i(N_2)_0)[63:32]$	$(q_i(N_2)_1)[63:32]$	$(q_i(N_2)_2)[63:32]$	• • •
時間 → C_{-1}	$Y[31:0]$	$Y[63:32]$	$Y[95:64]$	• • •

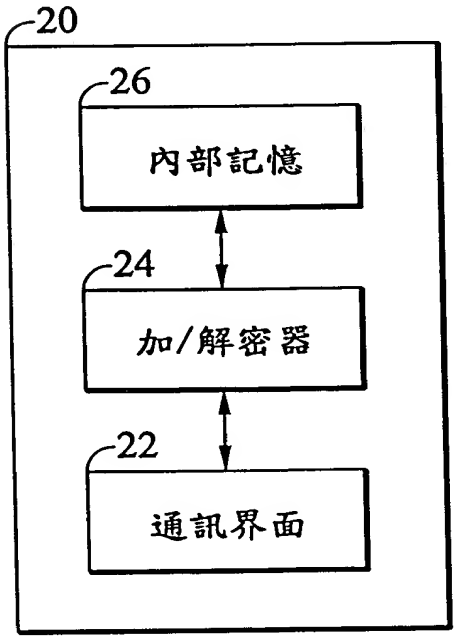
第 2 圖

	j = 0	j = 1	j = 2	• • •
	Y[31:0]	Y[63:32]	Y[95:64]	• • •
	$(a_i(B_0)[31:0]$	$(a_i(B_1)[31:0]$	$(a_i(B_2)[31:0]$	• • •
+		$(a_i(B_0)[63:32]$	$(a_i(B_1)[63:32]$	• • •
時間 →	$R_{i+1}[31:0]$	$R_{i+1}[63:32]$	$R_{i+1}[95:64]$	• • •

第 3 圖



第 4 圖



第 5 圖